

<<汇编语言>>

图书基本信息

书名：<<汇编语言>>

13位ISBN编号：9787302071952

10位ISBN编号：7302071950

出版时间：2003-9-1

出版时间：清华大学出版社

作者：王爽

页数：315

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<汇编语言>>

前言

汇编语言是很多相关课程(如：数据结构、操作系统、微机原理等)的重要基础。其实仅从课程关系的角度讨论汇编语言的重要性未免片面，概括地说，如果读者想从事计算机科学方面的工作的话，汇编语言的基础是必不可缺的。原因很简单，我们的工作平台、研究对象都是机器，汇编语言是人和计算机沟通的最直接的方式，它描述了机器最终所要执行的指令序列。我们想深入研究英国文化，不会英语行吗？

<<汇编语言>>

内容概要

《汇编语言》具有如下特点；采用全新的结构对课程的内容进行了组织，对知识进行最小化分割，为读者构造了循序渐进的学习线索；在深入本质的层面上对汇编语言进行讲解；对关键环节进行深入的剖析。

汇编语言是各种CPU所提供的机器指令的助记符的集合，人们可以用汇编语言直接控制硬件系统进行工作。

汇编语言是很多相关课程（如：数据结构、操作系统、微机原理等）的重要基础。

为了更好地引导、帮助读者学习汇编语言，作者以循序渐进的方式精心创作了《汇编语言》。

《汇编语言》可用作大学计算机专业本科生的汇编教材及希望深入学习计算机科学的读者的自学教材。

。

<<汇编语言>>

书籍目录

第1章 基础知识11.1 机器语言11.2 汇编语言的产生21.3 汇编语言的组成31.4 存储器31.5 指令和数据41.6 存储单元41.7 CPU对存储器的读写41.8 地址总线61.9 数据总线71.10 控制总线81.11 内存地址空间(概述)91.12 主板91.13 接口卡91.14 各类存储器芯片91.15 内存地址空间10第2章 寄存器(CPU工作原理)132.1 通用寄存器132.2 字在寄存器中的存储152.3 几条汇编指令162.4 物理地址182.5 16位结构的CPU182.6 8086CPU给出物理地址的方法192.7 “段地址×16+偏移地址=物理地址”的本质含义202.8 段的概念222.9 段寄存器232.10 CS和IP232.11 修改CS、IP的指令302.12 代码段32实验1 查看CPU和内存,用机器指令和汇编指令编程33第3章 寄存器(内存访问)453.1 内存中字的存储453.2 DS和[address]463.3 字的传送473.4 mov、add、sub指令493.5 数据段513.6 栈533.7 CPU提供的栈机制553.8 栈顶超界的问题583.9 push、pop指令603.10 栈段64实验2 用机器指令和汇编指令编程67第4章 第1个程序714.1 一个源程序从写出到执行的过程714.2 源程序724.3 编辑源程序774.4 编译774.5 连接794.6 以简化的方式进行编译和连接824.7 1.exe的执行834.8 可执行文件中的程序装入内存并运行的原理834.9 程序执行过程的跟踪85实验3 编程、编译、连接、跟踪88第5章 [bx]和loop指令895.1 [bx]915.2 Loop指令935.3 在Debug中跟踪用loop指令实现的循环程序965.4 Debug和汇编编译器Masm对指令的不同处理1025.5 loop和[bx]的联合应用1055.6 段前缀1085.7 一段安全的空间1095.8 段前缀的使用111实验4 [bx]和loop的使用113第6章 包含多个段的程序1146.1 在代码段中使用数据1146.2 在代码段中使用栈1186.3 将数据、代码、栈放入不同的段120实验5 编写、调试具有多个段的程序123第7章 更灵活的定位内存地址的方法1267.1 and和or指令1267.2 关于ASCII码1267.3 以字符形式给出的数据1277.4 大小写转换的问题1287.5 [bx+idata]1317.6 用[bx+idata]的方式进行数组的处理1327.7 SI和DI1347.8 [bx+si]和[bx+di]1367.9 [bx+si+idata]和[bx+di+idata]1387.10 不同的寻址方式的灵活应用139实验6 实践课程中的程序147第8章 数据处理的两个基本问题1488.1 bx、si、di、bp1488.2 机器指令处理的数据所在位置1498.3 汇编语言中数据位置的表达1508.4 寻址方式1518.5 指令要处理的数据有多长? 1528.6 寻址方式的综合应用1538.7 div指令1568.8 伪指令dd1588.9 dup159实验7 寻址方式在结构化数据访问中的应用160第9章 转移指令的原理1629.1 操作符offset1629.2 jmp指令1649.3 依据位移进行转移的jmp指令1649.4 转移的目的地址在指令中的jmp指令1689.5 转移地址在寄存器中的jmp指令1699.6 转移地址在内存中的jmp指令1699.7 jcxz指令1719.8 loop指令1729.9 根据位移进行转移的意义1739.10 编译器对转移位移超界的检测174实验8 分析一个奇怪的程序174实验9 根据材料编程175第10章 call和ret指令17810.1 ret和retf17810.2 call指令18010.3 依据位移进行转移的call指令18010.4 转移的目的地址在指令中的call指令18110.5 转移地址在寄存器中的call指令18210.6 转移地址在内存中的call指令18210.7 call和ret的配合使用18410.8 mul指令18710.9 模块化程序设计18810.10 参数和结果传递的问题18810.11 批量数据的传递19010.12 寄存器冲突的问题191实验10 编写子程序194课程设计1200第11章 标志寄存器20211.1 ZF标志20211.2 PF标志20311.3 SF标志20411.4 CF标志20511.5 OF标志20611.6 adc指令20811.7 sbb指令21111.8 cmp指令21111.9 检测比较结果的条件转移指令21511.10 DF标志和串传送指令21911.11 pushf和popf22311.12 标志寄存器在Debug中的表示223实验11 编写子程序224第12章 内中断22512.1 内中断的产生22512.2 中断处理程序22612.3 中断向量表22612.4 中断过程22712.5 中断处理程序22812.6 除法错误中断的处理22912.7 编程处理0号中断22912.8 安装23212.9 do023512.10 设置中断向量23812.11 单步中断23812.12 响应中断的特殊情况239实验12 编写0号中断的处理程序240第13章 int指令24113.1 int指令24113.2 编写供应用程序调用的中断例程24213.3 对int、iret和栈的深入理解24513.4 BIOS和DOS所提供的中断例程24713.5 BIOS和DOS中断例程的安装过程24813.6 BIOS中断例程应用24813.7 DOS中断例程应用250实验13 编写、应用中中断例程251第14章 端口25414.1 端口的读写25414.2 CMOS RAM芯片25514.3 shl和shr指令25614.4 CMOS RAM中存储的时间信息258实验14 访问CMOS RAM260第15章 外中断26115.1 接口芯片和端口26115.2 外中断信息26115.3 PC机键盘的处理过程26315.4 编写int 9中断例程26415.5 安装新的int 9中断例程272实验15 安装新的int 9中断例程274指令系统总结274第16章 直接定址表27616.1 描述了单元长度的标号27616.2 在其他段中使用

<<汇编语言>>

数据标号27816.3 直接定址表28116.4 程序入口地址的直接定址表285实验16 编写包含多个功能子程序的中断例程289第17章 使用BIOS进行键盘输入和磁盘读写29017.1 int 9中断例程对键盘输入的处理29017.2 使用int 16h中断例程读取键盘缓冲区29117.3 字符串的输入29417.4 应用int 13h中断例程对磁盘进行读写298实验17 编写包含多个功能子程序的中断例程301课程设计2302附注304附注1 Intel系列微处理器的三种工作模式304附注2 补码305附注3 汇编编译器(masm.exe)对jmp的相关处理307附注4 用栈传递参数310附注5 公式证明313

<<汇编语言>>

编辑推荐

《汇编语言》采用了独特的教学理念，完全按照循序渐进的原则写成。学习是一个动态的过程，《汇编语言》的重要的宗旨就是对学习的过程进行引导。因此，作者打破了传统教材按照静态的知识结构对内容进行组织的体例，在其中构造了一个贯穿始终的可以步步深入的学习线索，使得读者可以真正深入地理解机器运行程序的机理和获得高质量的底层编程体验，为计算机科学相关课程的学习和研究打下坚实的理论 and 实践基础。

采用全新的结构对课程内容进行了组织，对知识进行最小化分割，为读者构造了循序渐进的学习线索。

在深入本质的层面上对汇编语言进行讲解。
对关键环节进行深入的剖析。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>